

FILE: 6349USF.RT

APPARATUS AND METHOD FOR DATA PROCESSING USING MULTIPLY- ACCUMULATE INSTRUCTIONS

CROSS-REFERENCE TO RELATED APPLICATION

This application claims the priority benefit of provisional application serial no. 60/212,954, filed June 21, 2000, the full disclosure of which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field of Invention

The present invention relates to an apparatus and a method of data processing system. More particularly, the present invention relates to an apparatus and a method of data processing system that uses multiply-accumulate instructions. The data processing system can easily detect an overflow case and simplify calculations to save execution time.

Description of Related Art

In the field of data processing, it is necessary to be able to perform certain operations upon operands stored in various data registers. One such operation is to multiply an N-bit operand by a second N-bit operand and add a third N-bit operand to the result. Another similar operation is to multiply an N-bit operand by a second N-bit operand and add a 2N-bit operand to the result.

Figure 1 is a block diagram showing a conventional multiply-accumulator unit for a data processing system in US patent No. 5,583,804, titled "DATA PROCESSING

FILE: 6349USF.RTF

USING MULTIPLY-ACCUMULATE INSTRUCTIONS". This system is capable of performing a first class of multiply-accumulate instructions in the form of $N*N+2N \rightarrow 2N$, and a second class of multiply-accumulate instructions in the form of $N*N+N \rightarrow N$.

5 The multiply-accumulator unit comprises a first data register 10, a second data register 20, an $N*N$ multiplier 30, an $N+N$ accumulator 40, and a $2N+2N$ accumulator 50. The multiplier 30 is capable of calculating $N*N$ to get an N or a $2N$ result. The $N+N$ accumulator 40 is capable of calculating $N+N$ to get an N result. The $2N+2N$ accumulator 50 is capable of calculating $2N+2N$ to get a $2N$ result.

10 However, it is possible while performing an operation on an $N*N+N \rightarrow N$ class instruction, that the final N result is greater than can be represented in a result of N -bit size. It is important when this situation happens, that the user be made aware that an overflow condition has occurred in the operation. A disadvantage of the multiply-accumulator unit of Figure 1 is the unit's inability to show this overflow condition. To
15 provide this critical information to the user in as efficient manner as possible is one reason the multiply-accumulator unit of the present invention was developed.

SUMMARY OF THE INVENTION

Accordingly, an object of the present invention is to provide a data processing
20 system comprising a single accumulator and capable of detecting overflow conditions and capable of performing multiply-accumulate instructions. Hence, system architecture is simplified and at the same time, valuable overflow information is provided. The data processing system can easily detect an overflow case and simplify calculations to save execution time.

FILE: 6349USF.RTT

To achieve these and other advantages and in accordance with the purpose of the invention, as embodied and broadly described herein, the invention provides an apparatus for processing data, the apparatus comprising a first register bank of N-bit data processing registers, a second register bank of N-bit data processing registers, a selector, a multiplier and an accumulator. The selector is coupled to the first register bank and the second register bank and is used for selecting one of the first and second register banks and outputting N-bit data from the selected register banks. The outputted N-bit data and the N-bit data held in the second register bank form a 2N-bit addition operand. The multiplier is used for performing multiply operation upon a first operand and a second operand and outputting an 2N-bit multiplied result. The accumulator is coupled to the multiplier, the selector and the second register bank and is used for performing accumulate operation upon the 2N-bit multiplied result and the 2N-bit addition operand and outputting a 2N-bit accumulated result.

In the apparatus for processing data, the selector is further used for receiving a class signal, wherein the selector selects one of the first and second register banks in response to the class signal.

The class signal of the above-mentioned apparatus for processing data is used for indicating a first class of instruction or a second class of instruction.

The apparatus for processing data further comprises a detecting device, coupled to the accumulator, for receiving the 2N-bit accumulated result and for checking if a case of overflow occurs.

In the apparatus for processing data, the outputted N-bit data from the selector and the N-bit data held in the second register bank are formed in combination as a first N-bit part and a second N-bit part of the 2N-bit addition operand, the accumulated result

FILE: 6349USF.RTF

includes a third N-bit part and a forth N-bit part, when the class signal is the second class of instruction, the detecting device comparing the first N-bit part of the 2N-bit addition operand and the third N-bit part of the accumulated result to determine if the case of overflow occurs.

- 5 To achieve these and other advantages and in accordance with the purpose of the invention, as embodied and broadly described herein, the invention provides a method for processing data using an apparatus having a first register bank of N-bit data processing registers, a second register bank of N-bit data processing registers, a selector, a multiplier and an accumulator, the method comprising selecting one of the first and
- 10 second register banks and outputting N-bit data from the selected register banks, wherein the outputted N-bit data and the N-bit data held in the second register bank form a 2N-bit addition operand; performing multiply operation upon a first operand and a second operand and outputting an 2N-bit multiplied result; performing accumulate operation upon the 2N-bit multiplied result and the 2N-bit addition operand and
- 15 outputting a 2N-bit accumulated result.

In the above-mentioned method for processing data, the step of selecting one of the first and second register banks and outputting N-bit data from the selected register banks further comprising a step of receiving a class signal is determined by a class signal received by the selector.

- 20 The class signal is used for indicating a first class of instruction or a second class of instruction.

In the above-mentioned method for processing data further comprises a step of receiving the 2N-bit accumulated result and checking if a case of overflow occurs.

FILE: 6349USF.RT

In the above-mentioned method for processing data, the outputted N-bit data from the selector and the N-bit data held in the second register bank are formed in combination as a first N-bit part and a second N-bit part of the 2N-bit addition operand, the accumulated result includes a third N-bit part and a forth N-bit part, when the class
5 signal is the second class of instruction, comparing the first N-bit part of the 2N-bit addition operand and the third N-bit part of the accumulated result to determine if the case of overflow occurs.

It is to be understood that both the foregoing general description and the following detailed description are exemplary, and are intended to provide further
10 explanation of the invention as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings are included to provide a further understanding of the invention, and are incorporated in and constitute a part of this specification. The
15 drawings illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention. In the drawings,

Figure 1 is a block diagram showing a conventional multiply-accumulator unit for a data processing system; and

Figure 2 is a block diagram showing a multiply-accumulator unit for a data
20 processing system according to the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings. Wherever

FILE: 6349USF.RT

possible, the same reference numbers are used in the drawings and the description to refer to the same or like parts.

Refer to Figure 2, which is a block diagram showing a multiply-accumulator unit 200 for a data processing system according to the present invention. The multiply-accumulator unit 200 includes a selector 160, a multiplier 130, and an accumulator 150. Data are supplied from a special register bank 110 and a general register bank 120. The multiplier 130 is coupled to the accumulator 150. The special register bank 110 and the general register bank 120 are coupled to the selector 160. The selector 160 is coupled to the accumulator 150. The general register bank 120 is also directly coupled to the accumulator 150. The multiply-accumulator unit 200 further includes a detecting device 170 coupled to the accumulator 150.

The multiplier 130 is capable of multiplying two N-bit operands and then gets a 2N-bit multiplied result 134. For example, $N*N$ to get a 2N result. As shown in Fig.2, a first N-bit operand denoted as E and a second N-bit operand denoted as F are inputted into the multiplier 130. A 2N-bit multiplied result 134 is generated by the multiplier 130. The multiplied result 134 from multiplier 130 is sent to the 2N-bit accumulator 150 and is added by an addition operand. The addition operand is also 2N bits including a first N-bit part and a second N-bit part. In the invention, just one accumulator is necessary to provide more desired calculations. For example, in the prior art, if two calculations such as $N*N+N \rightarrow N$ and $N*N+2N \rightarrow 2N$ are desired, there are at least two accumulators are necessary to provide these desired calculations. In the architecture of the invention, only one accumulator is necessary for the two calculations. The circuits as shown in Fig. 2 can implement the feature.

FILE: 6349USF.RT

In the invention, a class signal 100 is provided to the selector 160 to indicate one of two different classes of instructions. The class signal 100 indicates which class of instructions is being operated on. For example, a first class of instructions such as $N*N+2N \rightarrow 2N$ or a second class of instructions such as $N*N+N \rightarrow N$. The class signal is set by a decoding instruction supplied to the multiply-accumulator unit 200. The first class of instruction needs more execution time and a more precise result is generated therefrom. The second class of instruction needs less execution time than the first class of instruction.

When the class of instructions is the first class, i.e., if the desired calculation is $N*N+2N \rightarrow 2N$, the class signal 100 causes the selector 160 to provide data from the general register bank 120 to the accumulator 150. That is, the 2N bits of the addition signal 152 includes (N, N). The first N-bit part of the addition signal 152 is denoted by C, and the second N-bit part of the addition signal 152 is denoted by D. The first N-bit part C is provided by the general register bank 120. The second N-bit part D is also provided from the general register bank 120, as shown in Fig.2.

When the class of instructions is the second class, i.e., $N*N+N \rightarrow N$, the class signal 100 causes the selector 160 to provide data from the special register bank 110 to the accumulator 150. That is, the special register 110 provides the first N-bit part C of the addition operand 152. The special register bank 110 can be accessed by users under software control. The second N-bit part D is provided from the general register bank 120. In the multiply-accumulator unit 200 of the preferred embodiment of the invention, a 2N-bit (152) will be added by the accumulator 150 and a 2N-bit accumulated result will be generated, no matter the calculation of $N*N+N \rightarrow N$ just needs a N-bit data to be added and a N-bit accumulated result to be generated. The

FILE: 6349USF.RT

architecture of the invention provides some advantages. For example, a case of overflow will be easily detected in such design. The details will be described below. Another advantage is simplifying some calculations to save execution time. For example, if a calculation such as $\sum_{k=0}^n X_k Y_k = X_0 Y_0 + X_1 Y_1 + \dots + X_n Y_n$ is desired, the

5 comparison between the preferred embodiment and the prior art is as followed:

In the prior art, the program language is:

```

for (k=0; k<=n; k++) {
    Move Xk to R0
    Move Yk to R1
10    R2=R0*R1 + R2; MLA R2, R0, R1, R2
    k=k+1
}

```

wherein the "MLA" is an instruction for N*N+N=N, the result after executing the "MLA" instruction is 32-bit in length.

15

In the preferred embodiment, the program language is:

```

for (k=0; k<=n; k++) {
    Move Xk to R0
    Move Yk to R1
20    (RCP, R2)=R0*R1 + (RCP, R2); MLA R2, R0, R1, R2
    k=k+1
}

```

wherein the "MLA" is an instruction for N*N+N=N, the result after executing the "MLA" instruction in the program of the invention is 64-bit in length.

FILE: 6349USF.RT

In the preferred embodiment of the invention, a result of summation is 64-bit and the prior art is 32-bit. If a 64-bit result is required in the prior art, a calculation of $N*N+2N \rightarrow 2N$ is necessary, which needs more execution time than that in the embodiment. That is, in the invention, a simplified calculation is necessary for the same as in the prior art.

After accumulation, the accumulator 150 generates an accumulated result 154. The accumulated result 154 is 2N-bit and includes a first N-bit part H and a second N-bit part I. The accumulated result 154 will be an output result of the multiply-accumulator unit 200. The accumulated result 154 also can be outputted to a detection device 170 to detect a case of overflow, which is indicated by a detective result 172.

In such a case that the class of instructions is the second class, i.e., calculation of $N*N+N \rightarrow N$ is desired. The detection device 170 will compare the first N-bit part H of the accumulated result 154 with the first N-bit part C of the addition operand 152, which is supplied by the special register bank 110. If the first N-bit part H of the accumulated result 154 is not the same as the first N-bit part C of the addition operand 152, it means that an overflow case is occurred in this calculation.

For clarity, the two N-bit operands that inputted into the multiplier 130 are respectively denoted as E and F. The calculation of $N*N+N \rightarrow N$ instruction can be implemented by the invention as $E*F+CD \rightarrow HI$. In this case that C is not equal to the H, that means that an overflow occurs. The calculation of $N*N+2N \rightarrow 2N$ instruction can also be implemented by the same architecture of the invention as $E*F+CD \rightarrow HI$.

For an $N*N+2N \rightarrow 2N$ class instruction, the accumulator adds CD to the result of the $E*F$ multiply operation, to get an accumulated HI result. Remembering that the accumulator only performs one type of calculation, $2N+2N \rightarrow 2N$. For an $N*N+N \rightarrow N$

FILE: 6349USF.R1

class instruction the accumulator adds CD to the result of the E*F multiply operation, to get an HI result. C, in this case, is provided by the special register bank 110. If after the addition operation H does not equal C, then there is an overflow.

This overflow indicator provides users with useful information in a quick and convenient manner whereas the multiply-accumulator unit of Figure 1 does not provide this overflow indication. This is another advantage of the present invention.

It will be apparent to those skilled in the art that various modifications and variations can be made to the structure of the present invention without departing from the scope or spirit of the invention. In view of the foregoing, it is intended that the present invention cover modifications and variations of this invention provided they fall within the scope of the following claims and their equivalents.